

REMARKS/ARGUMENTS

Claims 1-38 are pending in the present application.

This Amendment is in response to the Office Action mailed March 15, 2004. In the Office Action, the Examiner objected to the specification and the drawings, rejected claims 1-38 under 35 U.S.C. §112; and claims 1-38 under 35 U.S.C. §103(a). Applicant amended claims 13-15 and 28-30. Reconsideration in light of the amendments and remarks made herein is respectfully requested.

Drawings

The Examiner objects to the drawings because item 610 is not shown. In response, Applicant has amended the specification paragraph [0055] to remove the reference to “the guard 610”. In addition, the items 645 and 655 in Figure 6 are not mentioned in the description. Applicant has amended the Specification, paragraphs [0055] and [0056].

Accordingly, Applicant respectfully requests the objection to the drawings be withdrawn.

Specification

1. In the Office Action, the Examiner objects to the Specification because there is lack of cross-references to related applications. In response, Applicant has elected not to cross-reference any applications in this application.
2. The Examiner states that the trademark JAVA is used and it should be capitalized. In response, Applicant has amended the Specification to capitalized all appearances of the trademark JAVA.

Accordingly, Applicant respectfully requests the objection to the Specification be withdrawn.

Rejection Under 35 U.S.C. § 112

1. Trademark in claims

The Examiner states that claims 13-15, 28-30, contain the trademark/tradename JAVA/JVMDI and therefore do not comply with requirements of 35 U.S.C. §112, second

paragraph, citing Ex parte Simpson, 218 USPQ 1020 (Bd. App. 1982). Applicant respectfully disagrees.

The presence of a trademark or tradename in a claim is not, per se, improper under 35 U.S.C. §112, second paragraph. MPEP 2173.05(a). "...the use of trademarks having definite meanings is permissible in patent applications" MPEP 608.01(v).

In Ex parte Simpson and Roberts, the appellant use the trademark Hypalon in the specification and claims. The court sustained the Examiner's rejection on the basis of indefiniteness. However, the court's reasoning was not based on the mere fact that Hypalon is a trademark, but based on the uncertain claim scope. The Court stated,

"The claim scope is uncertain as regards the material which forms the 'Hypalon' membrane. On the one hand, the claim language may be very narrowly construed to a particular chlorosulphonated ethylene having a specific group of additives employed by the owner of the 'Hypalon' trademark to produce the desired properties, or on the other hand the claim language might be asserted by appellants,..., to broadly encompass every synthetic resin..."(Ex parte Simpson at 1024, 1022).

Here, JAVA is used as an adjective, not as a noun, to definitely characterize a particular implementation of a function (JAVA method), a field (JAVA field), a virtual machine (JAVA virtual machine), and a debug interface (JAVA virtual machine debug interface). There is no confusion or indefinite regarding the JAVA method, JAVA field, JAVA virtual machine, and the JVM debug interface. Furthermore, JAVA is used in the technical field to indicate a high-level programming language. The use of the JAVA programming language has been so widespread the there is no indefiniteness or confusion.

Accordingly, Applicant respectfully requests the rejection to claims 13-15, 28-30 under 35 U.S.C. §112, second paragraph be withdrawn.

Rejection Under 35 U.S.C. § 103

1. In the Office Action, the Examiner rejected claims 1-38 under 35 U.S.C. §103(a) as being unpatentable over U.S. Pre Grant Publication 2001/0047510 issued to Angel et al. ("Angel") in view of "Poor Man's Watchpoints", by Max Copperman and Jeff Thomas (1995)

(“Copperman”). Applicant respectfully traverses the rejection and contend that the Examiner has not met the burden of establishing a prima facie case of obviousness.

Angel discloses a byte code instrumentation. A technique to instrument a byte code program includes examining the byte code, selecting portions of the byte code for instrumentation, and instrumenting the portions to provide instrumented byte code (Angel, paragraph [0014]). Memory access instructions are instrumented to detect illegal memory operations at runtime (Angel, paragraph [0091]). In addition, exiting and entering blocks of code where variables become defined and undefined are monitored (Angel, paragraph [0091]).

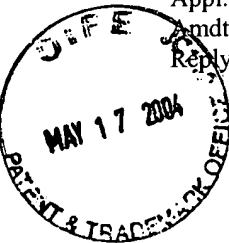
Copperman discloses a technique to implement watch points using code patching. When the user sets a watch point, the debugger sets the register \$fp to point to a register save area in the debuggee’s static data space. When no watch points are set, the first instruction in the patch branches around the rest of the patch if \$fp contains (Copperman, page 38, third paragraph under section “The Debuggee”).

Angel and Copperman, taken alone or in any combination, does not disclose, suggest, or render obvious (1) compiling a function including a byte code sequence, (2) generating an instrumentation code corresponding to a field watch, (3) guarding execution of the instrumentation code if the field watch is not activated; and (4) inserting the instrumentation code to the native code. There is no motivation to combine Angel and Copperman because neither of them addresses the problem of compilation according to a field watch. There is no teaching or suggestion that guarding execution of instrumentation code or a field byte code accessing or modifying the field is present. Angel, read as a whole, does not suggest the desirability of generating an instrumentation code corresponding to the field watch.

The Examiner states that Angel discloses, generating an instrumentation code corresponding to a field watch of the field (Office Action, page 6). Applicant respectfully disagree. The cited paragraph, [0125], merely states that “[d]escribed below are methods of automatically editing the executable byte code representation...for generating instrumented byte code” (Angel, page 11, paragraph [0125]). There is no teaching or suggestion on the use of a field watch in generating instrumentation code. A field watch sequence may include instruction sequence to spill the mimic stack operands, which are live at the field access point, to their canonical spill locations (See, for example, Specification, page 14, paragraph [0047]).

The Examiner further states that Copperman discloses guarding execution of the instrumentation code if the field watch is not activated by disclosing setting or not setting the watchpoints, or entering or enabling a watchpoint command (Office Action, page 7). Applicant respectfully disagrees. A watchpoint or watchpoint command is not the same as a field watch as discussed above.

Therefore, Applicant believes that independent claims 1, 16, 31 and their respective dependent claims are distinguishable over the cited prior art references. Accordingly, Applicant respectfully requests the rejections under 35 U.S.C. §112 and 35 U.S.C. §103(a) be withdrawn.



Conclusion

Applicant respectfully requests that a timely Notice of Allowance be issued in this case.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Dated: May 13, 2004

By

Thinh V. Nguyen

Reg. No. 42,034

Tel.: (714) 557-3800 (Pacific Coast)

RECEIVED

MAY 20 2004

12400 Wilshire Boulevard, Seventh Floor
Los Angeles, California 90025

Technology Center 2100

CERTIFICATE OF MAILING/TRANSMISSION (37 CFR 1.8A)

I hereby certify that this correspondence is, on the date shown below, being:

MAILING

FACSIMILE

☒ deposited with the United States Postal Service
as first class mail in an envelope addressed to:
Commissioner for Patents, PO Box 1450,
Alexandria, VA 22313-1450.

☐ transmitted by facsimile to the Patent and
Trademark Office.

Date: May 13, 2004


Tu Nguyen

May 13, 2004

Date